

**AMENDMENTS TO THE CLAIMS**

1-40. (Cancelled)

41. (Previously Presented) A method in a computer system for accessing a collection of data items, the method comprising:

defining a pointer to indicate a location for a bucket;

defining a lower bound to indicate a number of items in the collection; and

determining based on the lower bound whether the collection has a data item;

and

when it is determined that the collection has a data item,

adjusting the lower bound to indicate that the collection has one less data item;

reading from the bucket at the location defined by the pointer using a synchronization access mode of sync so that if the bucket is currently marked empty, the reading is delayed until the bucket is marked full and so that the bucket is marked empty upon reading to prevent subsequent reading from the bucket until the bucket is written to;

removing a data item from association with the bucket at the location defined by the pointer; and

writing to the bucket at the location defined by the pointer using a synchronization access mode of sync so that the bucket is marked full allowing reading from the bucket;

when access to the location by a thread is blocked,

enabling an exception to be raised when the location is next accessed;

and

blocking the thread; and

when an exception is raised as a result of access by another thread to that location,  
completing the access by that other thread to that location; and  
restarting execution of the blocked thread.

42. (Original) The method of claim 41 wherein when access by the thread to the location is blocked, saving a state of the thread and storing a reference to the thread in the location.

43. (Original) The method of claim 42 wherein the reference is a pointer to a data structure that identifies the blocked thread and the saved state.

44. (Original) The method of claim 43 wherein the data structure indicates the value that was stored in the location before storing the reference.

45-53. (Cancelled)

54. (Previously Presented) A computer system for accessing data, comprising:

- a collection of buckets;
- a read counter and a write counter that point to buckets within the collection;
- a data structure for each bucket with locations for holding data at each of the buckets;
- a lower bound indicating whether a data item is currently stored in the data structure;
- a component that adjusts the lower bound to indicate a change in the number of data items;
- a component that accesses the bucket at the location defined by the pointer using a synchronization access mode of sync so that if the bucket is

currently marked empty, the reading is delayed until the bucket is marked full and so that the bucket is marked empty upon reading to prevent subsequent reading from the bucket until the bucket is written to;  
a component that modifies data items associated with the bucket at the location defined by the pointer; and  
a component that writes to the bucket at the location defined by the pointer using a synchronization access mode of sync so that the bucket is marked full allowing reading from the bucket;  
when access by a thread to a bucket is blocked,  
enabling an exception to be raised when the bucket is next accessed; and  
blocking the thread; and  
when an exception is raised as a result of access by another thread to that bucket,  
completing the access by that other thread to that bucket; and  
restarting execution of the blocked thread.

55-59. (Cancelled)

60. (New) The method of claim 41 wherein the collection of data items is stored in an array data structure.

61. (New) The method of claim 41 wherein the collection of data items is stored in a linked list data structure.

62. (New) The method of claim 41 wherein the collection of data items is stored in a tree data structure.

63. (New) The method of claim 41 wherein it cannot be ensured that the collection contains a data item, indicating that a data item cannot be removed.

64. (New) The method of claim 41 wherein the data items of the collection are accessed by multiple readers and writers.

65. (New) The method of claim 64 wherein the multiple readers and writers can access data items of different buckets simultaneously.

66. (New) The method of claim 41 wherein the data items of the collection are accessed by multiple producers.

67. (New) The method of claim 41 wherein the data items of the collection are accessed by multiple consumers.

68. (New) The method of claim 41 wherein the data is accessed by multiple reading threads and writing threads.

69. (New) The method of claim 68 wherein the data is accessed by multiple reading threads and writing threads simultaneously.

70. (New) The computer system of claim 54 wherein the data is stored in an array data structure.

71. (New) The computer system of claim 54 wherein the data is stored in a linked list data structure.

72. (New) The computer system of claim 54 wherein the data is stored in a tree data structure.

73. (New) The computer system of claim 54 wherein each bucket contains a pointer to a linked list of data items.

74. (New) The computer system of claim 54 wherein the data is accessed by multiple readers and writers.

75. (New) The computer system of claim 75 wherein the multiple readers and writers can access data of different buckets simultaneously.

76. (New) The computer system of claim 54 wherein the data is accessed by multiple producers.

77. (New) The computer system of claim 54 wherein the data is accessed by multiple consumers.

78. (New) The computer system of claim 54 wherein multiple producers and multiple consumers access the data simultaneously.